L3: Entry 52 of 97 File: USPT May 1, 2001

DOCUMENT-IDENTIFIER: US 6226744 B1
TITLE: Method and apparatus for authenticating users on a network using a smart card

Brief Summary Text (7):
Recent technological advances in the networking industry, however, have created a movement back to the original concept of shared computer resources. The emergence of high-speed communication networks gives users the same level of convenience when accessing computer information stored at a location remote from the user as when the information is stored at the user's own personal computer. Thus, a user has the advantage of utilizing the computing resources of their personal computer, while also having the benefit of connecting to a network having a wide variety of computing resources attached to it, such as powerful servers having high-speed processors and high-capacity mass storage devices, laser printers, and so forth. Further, a user was not limited to the information stored on their own computer, but could gain access to information stored on hundreds, even thousands, of individual computers linked together by a single network. An example of such networks are the Internet and World Wide Web (WWW).

Brief Summary Text (8):
Consequently, the popularity of the Internet and WWW is increasing at a phenomenal rate due to the fact that these networks provide a user with tremendous computing resources and information. A problem that has consistently plagued the networking industry in general, and the Internet and WWW in particular, however, is the authentication of network users. Invariably, some computers connected to the Internet and WWW contain restricted information that is accessible to only a limited number of users. As a result, it becomes necessary to confirm the identify of a user, and that the user has authorization to access the restricted information. Since the restricted information is stored remotely from the user, the authentication of the user to the access control agent responsible for the security of the restricted information requires an exchange of messages that constitute a user authentication protocol. The authentication protocol permits a user to prove his or her identity to the authentication server (AS) by demonstrating his or her knowledge of a secret, e.g. an access code such as a password or personal identification number (PIN), that is shared with the AS.

Brief Summary Text (11):
A practical mechanism for recovering strong cryptographic keys using weak secrets without exposure is provided through the use of smart cards. A smart card is a device that is typically the size of a credit card, having a microprocessor and limited storage memory. An example of a smart card is the Cryptoflex(.TM.) smart card by Schlumberger Electronic Transactions. Since a smart card has memory, a smart card can store a strong cryptographic key that is randomly chosen out of the total key space of the cryptographic algorithm in use. The probability of success with a brute force attack based on exhaustive search in the key space becomes negligible due to the strong key. Although the user must typically activate the smart card operation by authenticating himself using a weak initial secret, this interaction takes place directly between the user and the card without any involvement of untrusted media. Thereafter, all data exchanged over the untrusted network is sent under the protection afforded by encryption using the smart card's strong secret. Since the card is a simple device (not unlike a calculator), it is trusted by the principals involved.

Brief Summary Text (14):
In view of the foregoing, it can be appreciated that a substantial need exists for a method and apparatus for securing network computers having restricted information with smart cards without having to install hardware or software on the client.

Detailed Description Text (3):
This embodiment of the invention secures network computers, such as web sites, with smart cards without having to have a user or technician physically go to the PC and install hardware or software on the PC as in conventional authentication systems. Further, this embodiment of the invention permits a user to access and modify user information stored on the issued smart card.

Detailed Description Text (4):
This embodiment of the invention utilizes Cryptoflex smart cards and the Smarty reader. Cryptoflex smart cards are distributed through the mail after setting some initial values for access, e.g., PINs. A user inserts the card into the Smarty, and inserts the Smarty into a 3.5" floppy disk drive of a computer. When the user attempts to access a secure web site, a program is downloaded to the user's computer. The program allows the web site to access the floppy drive that contains the Smarty and smart card. Information from the card is accessed using the program and a PIN, and is compared with server information. Access to the web site will be either allowed or denied based upon the results of the comparison.

Detailed Description Text (15):
FIG. 3 is a block flow diagram of steps performed in accordance with one embodiment of the invention. As shown in FIG. 3, a Certified Authority (CA) distributes smart card 10 to a user at step 50. Smart card 10 stores user information provided by the CA, such as tokens, digital signatures, certificates, tickets, PIN, human resources identification number, and so forth, or personal information provided by the user such as a social security number, birth date, mother's maiden name, etc. Smart card 10 also performs data encryption and decryption functions, stores DES secret keys and digital certificates, and will generate and store public and private RSA cryptographic key pairs. Smart card 10 has an on-board math co-processor that performs the key generation and encryption/decryption calculations.

Detailed Description Text (18):
Client computer 14 uses a web browser to access secure gateway server 18 via WWW 16 at step 56. Secure gateway server 18 initiates authentication of the user of smart card 10 using authentication module 32. Authentication module 32 determines whether smart card 10 is present in client terminal 14. If smart card 10 is present in client terminal 14, then authentication module 32 initiates a download of a smart card interface module to client terminal 14 at step 58. The smart card interface module utilizes Active.TM. controls provided by Microsoft Corporation to create a set of controls for secure gateway server 18 to read and write information to smart card 10 using the API provided by Fischer. The smart card interface module will be described in more detail with reference to FIG. 4.

Detailed Description Text (19):
Once the smart card interface module is downloaded to client terminal 14, client terminal 14 executes the smart card interface module. The smart card interface module begins by modifying parameters for the operating system of client terminal 14 so that the operating system is aware of the software needed to interface with the smart card. The smart card interface module then requests a PIN to access smart card 10 at step 60. Authentication module 32 uses the smart card interface module and the PIN to access and read/write user information from/to smart card 10 at step 62.

Detailed Description Text (27):
In this embodiment of the invention, Visual C++ creates a framework for building an ActiveX control. Specific program code segments are then inserted into this framework that are application specific. The specific program code segments for a file "storessnctl.cpp" are listed below. Storessnctl.cpp is created to store key information. The control subclasses the default pushbutton control. A Fischer API header "sos.h" is also listed below, and a Fischer library "sdosm932.lib" is linked into the executable file when the specific program code is actually compiled, thereby permitting call functions such as SOSr_card_reset. The Fischer API's are, of

course, well-known in the art and therefore will not be described any further.

Detailed Description Text (28):
The beginning of storessnctl.ccp contains header infomation, event maps, property pages, and constructors/destructors.

Detailed Description Text (35):
The smart card is accessed at step 78 using the key received from the user at step 72. This example uses a default key to unlock the smart card:

Detailed Description Paragraph Table (1):
```
// StoressnCtl.cpp : Implementation of the CStoressnCtrl ActiveX Control class.
#include "stdafx.h" #include "storessn.h" #include "StoressnCtl.h" #include
"StoressnPpg.h" #include "EditDialog.h" #include "EditSsn.h" #include "sos.h"
#include "wininet.h" #include "afxinet.h" #ifdef _DEBUG #define new DEBUG_NEW #undef
THIS_FILE static char THIS_FILE[] = .sub.---- FILE.sub.---- ; #endif
IMPLEMENT_DYNCREATE(CStoressnCtrl, COleControl) // Message map
BEGIN_MESSAGE_MAP(CStoressnCtrl, COleControl) //{{(AFX_MSG_MAP(CStoressnCtrl)
//}}AFX_MSG_MAP ON_MESSAGE(OCM_COMMAND, OnOcmCommand)
ON_OLEVERB(AFX_IDS_VERB_PROPERTIES, OnProperties) END_MESSAGE_MAP() // Dispatch map
BEGIN_DISPATCH_MAP(CStoressnCtrl, COleControl) //{{AFX_DISPATCH_MAP (CStoressnCtrl)
DISP_STOCKPROP_CAPTION() //}}AFX_DISPATCH_MAP DISP_FUNCTION_ID(CStoressnCtrl,
"AboutBox", DISPID_ABOUTBOX, AboutBox, VT_EMPTY, VTS_NONE) END_DISPATCH_MAP() //
Event_map BEGIN_EVENT_MAP(CStoressnCtrl, COleControl)
//{{AFX_EVENT_MAP(CStoressnCtrl) //}}AFX_EVENT_MAP END_EVENT_MAP() // Property pages
BEGIN_PROPPAGEIDS (CStoressnCtrl, 1) PROPPAGEID (CStoressnPropPage::guid)
END_PROPPAGEIDS (CStoressnCtrl) // Initialize class factory and guide
IMPLEMENT_OLECREATE_EX(CStoressnCtrl, "STORESSN.StoressnCtrl. 1", 0xac78f35a,
0xe690, 0x11d0, 0xa0, 0xe9, 0, 0, 0xc0, 0x99, 0xbc, 0xc8) // Type library ID and
version IMPLEMENT_OLETYPELIB(CStoressnCtrl, _tlid, _wVerMajor, _wVerMinor) //
Interface IDs const IID BASED_CODE IID_DStoressn = { 0xac78f358, 0xe690, 0x11d0, {
0xa0, 0xe9, 0, 0, 0xc0, 0x99, 0xbc, 0xc8 } }; const IID BASED_CODE
IID_DStoressnEvents = { 0xac78f359, 0xe690, 0x11d0, { 0xa0, 0xe9, 0, 0, 0xc0, 0x99,
0xbc, 0xc8 } }; // Control type informnation static const DWORD BASED_CODE
_dwStoressnOleMisc = OLEMISC_ACTIVATEWHENVISIBLE .vertline.
OLEMISC_SETCLIENTSITEFIRST .vertline. OLEMISC_INSIDEOUT .vertline.
OLEMISC_CANTLINKINSIDE .vertline. OLEMISC_RECOMPOSEONRESIZE;
IMPLEMENT_OLECTLTYPE(CStoressnCtrl, IDS_STORESSN, _dwStoressnOleMisc) //
CStoressnCtrl::CStoressnCtrlFactory::UpdateRegistry - // Adds or removes system
registry entries for CStoressnCtrl BOOL
CStoressnCtrl::CStoressnCtrlFactory::UpdateRegistry(BOOL bRegister) { if (bRegister)
return AfxOleRegisterControlClass( AfxGetInstanceHandle (), m_clsid, m_lpszProgID,
IDS_STORESSN, IDB_STORESSN, afxRegApartmentThreading, _dwStoressnOleMisc, _tlid,
_wVerMajor, _wVerMinor); else return AfxOleUnregisterClass(m_clsid, m_lpszProgID); }
// CStoressnCtrl::CStoressnCtrl - Constructor CStoressnCtrl::CStoressnCtrl() {
InitializeIIDs(&IID_DStoressn, &IID_DStoressnEvents); } //
CStoressnCtrl::.sup..about. CStoressnCtrl - Destructor CStoressnCtrl::.sup..about.
CStoressnCtrl()
```